...e costruisci il tuo **LABORATORIO DIGITALE**

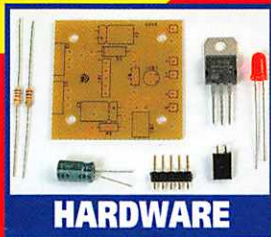
6,90 €

49

HARDWARE

DIGITALE DI BASE

MICROCONTROLLER

DIGITALE AVANZATO

TOTALMENTE PROGRAMMABILE!!!

9 771824 363008 41049

Peruzzo & C.

IN REGALO in questo fascicolo

1 Circuito stampato DG18
1 Circuito integrato 7805
1 Connettore maschio da c.s. a 5 vie diritto
1 Connettore femmina da c.s. a 2 vie a 90°
1 Condensatore 100 µF elettrolitico
1 Resistenza 5K6 5% 1/4 W
1 Resistenza 1K 5% 1/4 W
1 Diodo LED rosso

IN REGALO nel prossimo fascicolo

1 Circuito integrato 7805
1 Connettore femmina da c.s. a 2 vie a 90°
1 Condensatore 100 µF elettrolitico
1 Resistenza 1 K 5% 1/4 W
1 Diodo 1N4148 o 1N4007
4 Viti

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede. **Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail:** elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

"ELETTRONICA DIGITALE"
si compone di
70 fascicoli settimanali
da suddividere
in 2 raccoglitori.

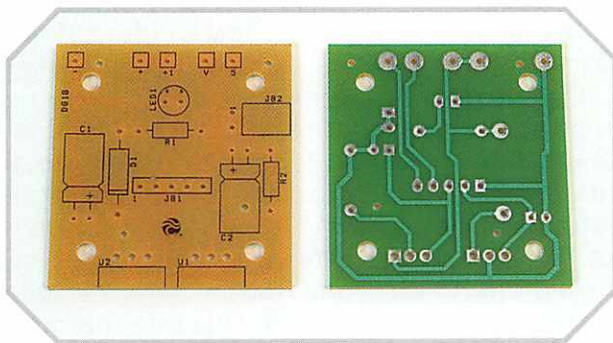
RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

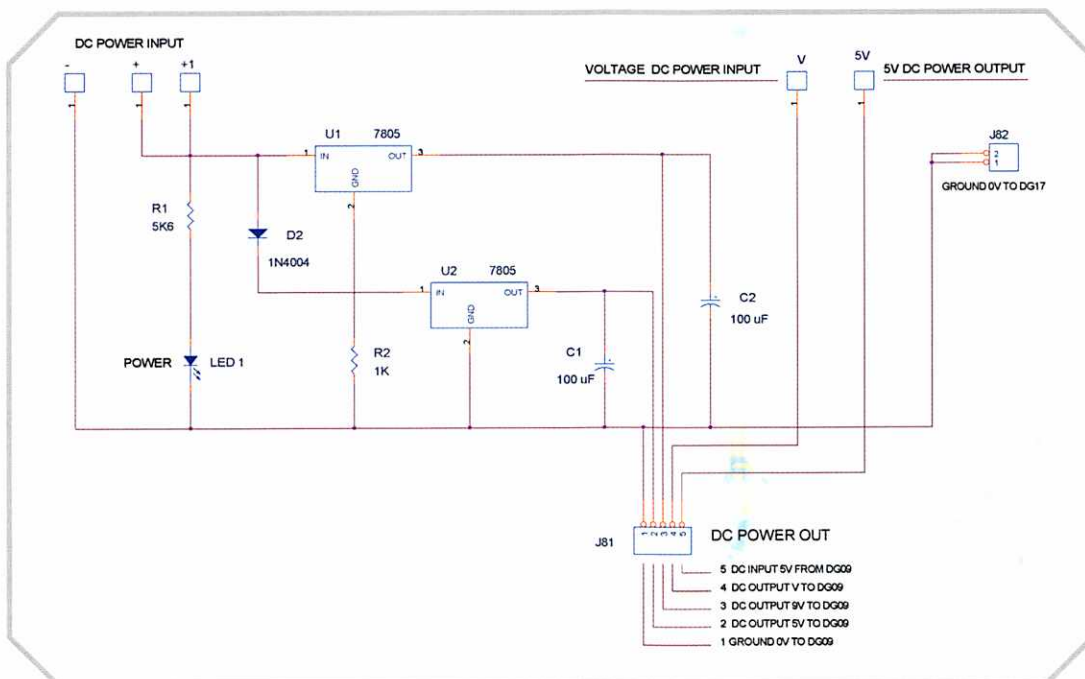
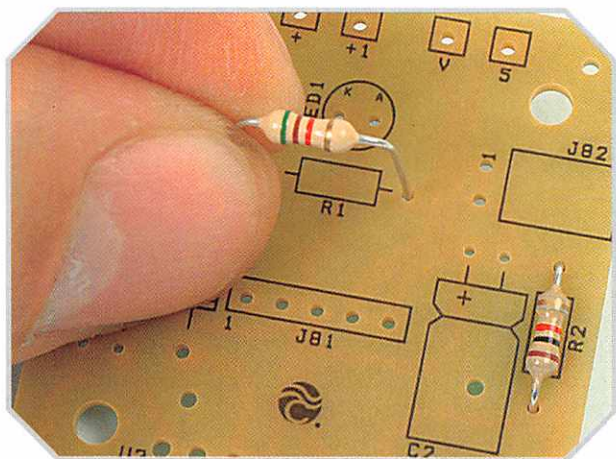
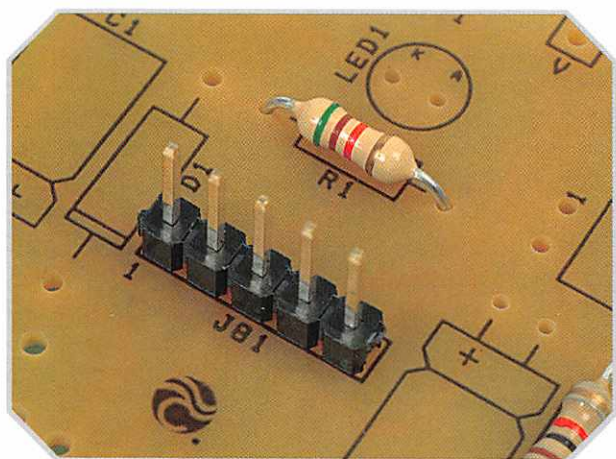
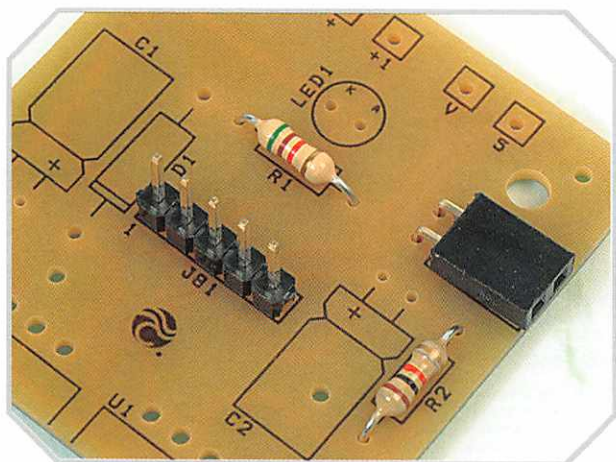
Il regolatore di tensione

Componenti allegati a questo fascicolo.

Circuito stampato DG18.

Con questo fascicolo vengono forniti il circuito stampato DG18 e alcuni componenti che monteremo sullo stesso.

Questo circuito stampato ha come funzione principale il supporto dei regolatori principali, che ci permetteranno di ottenere le due tensioni stabilizzate da 5 e 10 V, utilizzate sul laboratorio. Quando usiamo le batterie le tensioni sono nominalmente 4,5 e 9 V.

Descrizione

Questo circuito riceve l'energia sotto forma di tensione continua non regolata, tramite la scheda di filtro DG19 e genera a partire da essa le tensioni stabilizzate da 5 e 10 V che trasferisce al pannello principale del laboratorio tramite il connettore J81 e un cavetto a cinque fili già montato. Questo connettore si utilizza anche per portare la tensione dell'alimentatore variabile che passa tramite il terminale V e distribuisce al resto dei circuiti del pannello superiore il riferimento di alimentazione 0 V e il positivo dell'alimentazione da 5

Schema elettrico del circuito regolatore.

Montaggio di R1 da 5K6 e di R2 da 1 K.

Montaggio del connettore J81.

Scheda DG19 con i sui due connettori.

V, che ritornano attraverso la scheda di distribuzione DG09 situata sul pannello inferiore.

Lo schema

Se osserviamo lo schema e guardiamo i suoi terminali di ingresso siglati come (-) e (+), vedremo che corrispondono all'ingresso dell'alimentazione non stabilizzata, terminali negativo e positivo rispettivamente.

Il primo circuito che troviamo è composto da una resistenza da 5K6 e da un LED, il LED 1, che si utilizza come indicatore di ingresso dell'alimentazione.

Più avanti la corrente si divide in due rami, ognuno dei quali conduce la corrente a un regolatore, da un lato all'ingresso di U1 e dall'altro a U2 tramite un diodo raddrizzatore, D1.

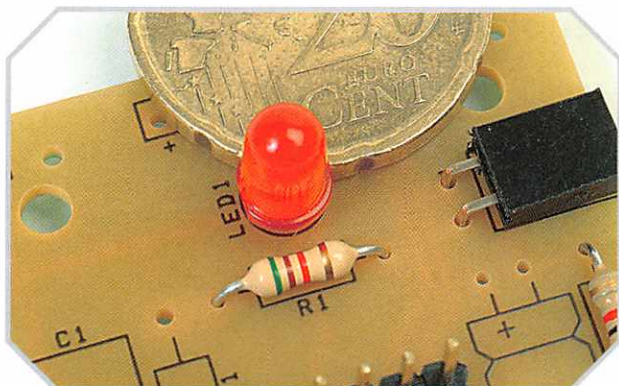
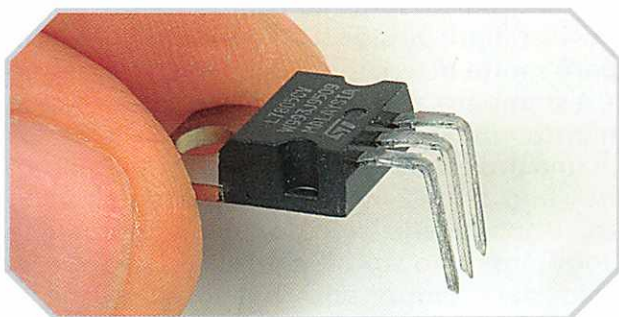
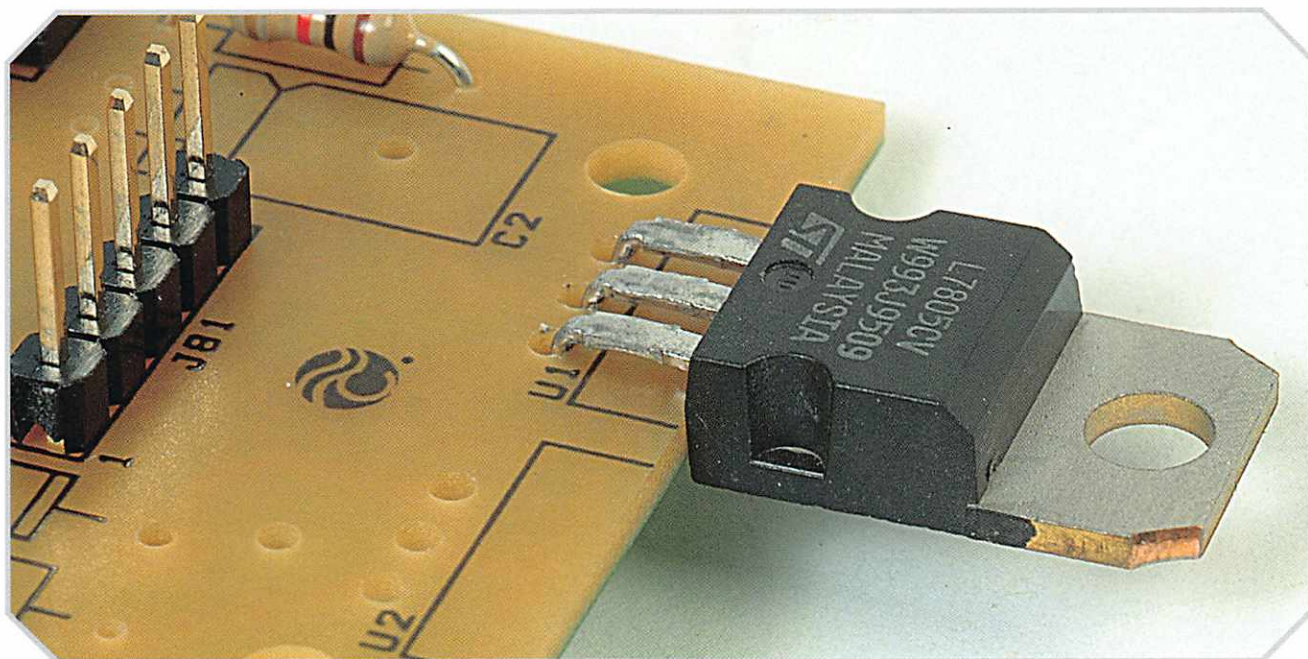
L'uscita del regolatore U2 è di 5 V, dato che si tratta di un regolatore da 5 V, il cui terminale di riferimento è collegato direttamente al negativo dell'alimentazione. Il secondo regolatore, anch'esso da 5 V, fornisce sulla sua uscita 10 V, dato che è stata inserita sul suo terminale di riferimento una resistenza da 1 K sulla quale circola una corrente costante, permettendoci di ottenere una tensione stabilizzata sulla sua uscita. Incrementando il valore di questa resistenza si aumenta la tensione di uscita e viceversa. I condensatori C1 e C2 filtrano la tensione di uscita di ogni regolatore.

Le uscite di questo circuito sono disponibili sul connettore a cinque vie J81, con accesso tramite l'esterno del pannello frontale e sono:

Terminale 1, riferimento di tensione 0 V.
Terminale 2, uscita da 5 V.
Terminale 3, uscita da 10 V.
Terminale 4, uscita di tensione V.

Su questo connettore si ricevono anche i 5 V tramite il pannello inferiore del laboratorio che corrisponde a:

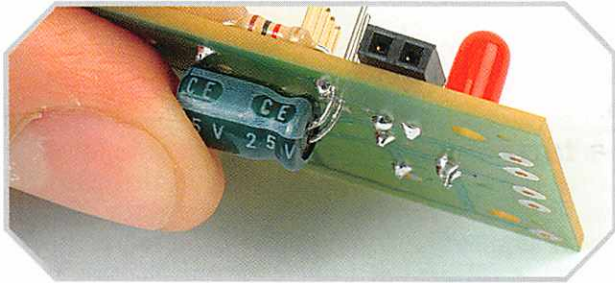
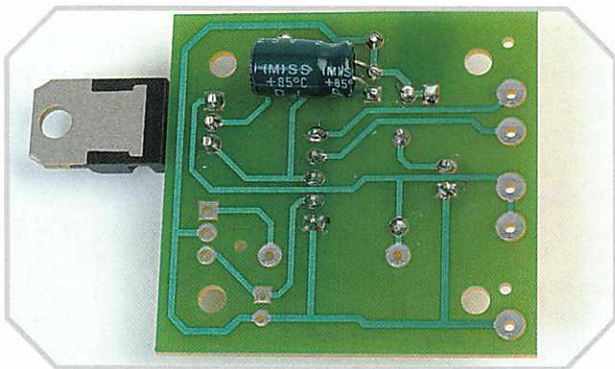
Terminale 5, ingresso di tensione da 5 V.

Il connettore J82 si utilizza per trasmettere il riferimento di tensione, cioè 0 V, al resto delle schede posizionate sul pannello superiore del laboratorio.

Il diodo LED deve essere saldato con la polarità corretta e rialzato di circa 2 mm.

Dettaglio del regolatore con i piedini piegati.

Montaggio

Il montaggio dei componenti su questo circuito stampato è piuttosto originale, in quanto l'altezza dei componenti è limitata dall'altezza a cui bisogna posizionare il connettore J81 da cinque vie. I primi due componenti da montare sono le resistenze R1 da 5K6 (verde, azzurro, rosso) e R2 da 1 K (marrone, nero, rosso) che dovranno essere fissate come d'abitudine, tagliando poi la parte in eccesso del reoforo.

Connettori

Continuiamo il montaggio con il connettore a cinque vie J81, che deve risultare perfettamente verticale. A questo scopo salderemo solamente un terminale e verificheremo che la sua posizione sia perfettamente verticale, in caso contrario correggeremo la posizione riscaldando e fondendo nuovamente lo stagno fino a quando non sarà posizionato correttamente. Dopo aver ottenuto questo obiettivo potremo saldare il resto dei terminali. Il connettore J82 si fissa di modo che il suo contenitore rimanga ben appoggiato sul circuito stampato. Inseriremo poi i terminali del diodo LED, tenendo presente la sua polarità, la zona piatta sul suo contenitore è quella più vicina al terminale del catodo.

Regolatore U1 saldato.

Il condensatore C2 si monta sul lato delle saldature, attenzione alla sua polarità!

Circuito DG19 con il materiale disponibile installato.

Regolatore

Prima di montare il regolatore è necessario conformarne i terminali piegandoli a 90°, come mostrato nelle immagini, poi lo salderemo nella posizione U1. Se per errore lo saldassimo nella posizione U2, non farebbe differenza, dato che l'altro regolatore avrà le stesse caratteristiche e verrà montato nello stesso modo e con la stessa conformazione dei terminali.

Condensatore

I condensatori elettrolitici C1 e C2 hanno un diametro che normalmente impedisce il loro montaggio sul lato dei componenti, vengono installati quindi sul lato rame, facendo però molta attenzione alle polarità. Il segno + è stampato anche sul rame, in questo caso monteremo il condensatore siglato come C2, in una posizione tale che i suoi terminali fuoriescano di soli 2 mm dall'altro lato del circuito. Dopo averlo saldato lo piegheremo e aggiungeremo anche un po' di stagno dal lato dei componenti per migliorarne il fissaggio.

Vista generale del laboratorio.

Indicatore di bassa temperatura

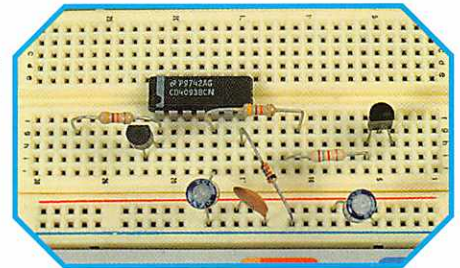
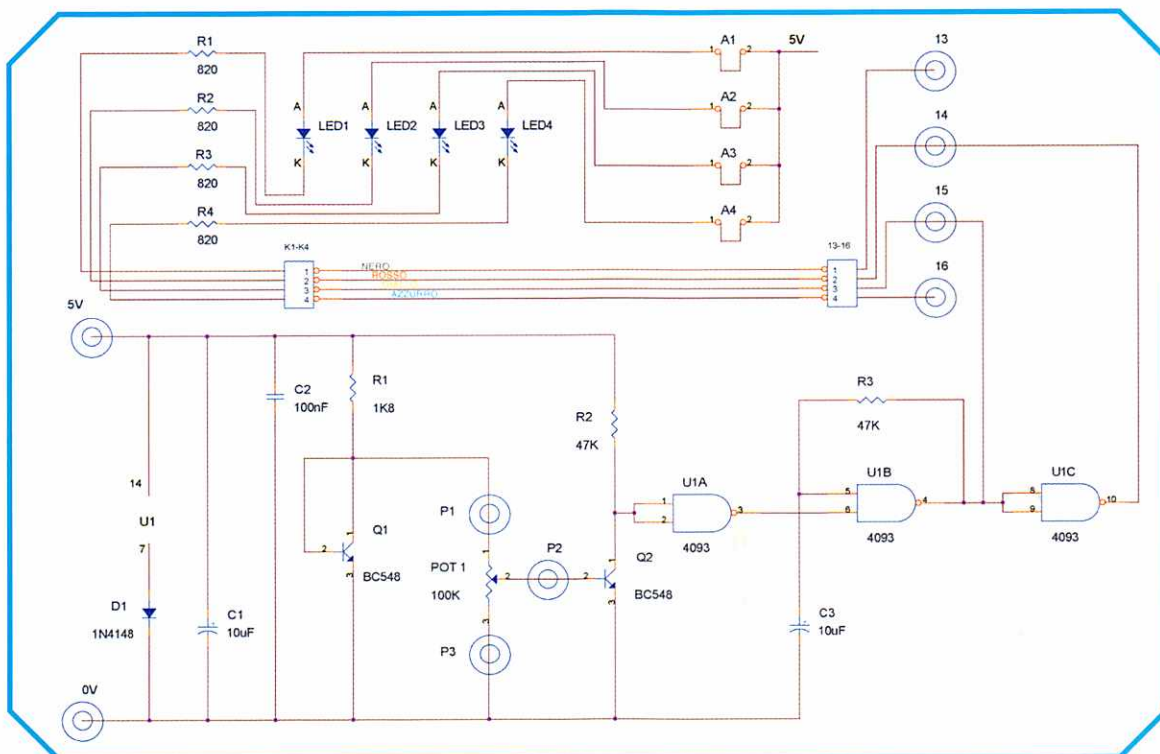
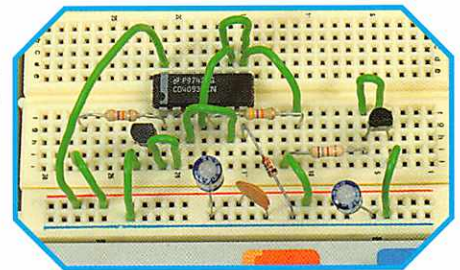
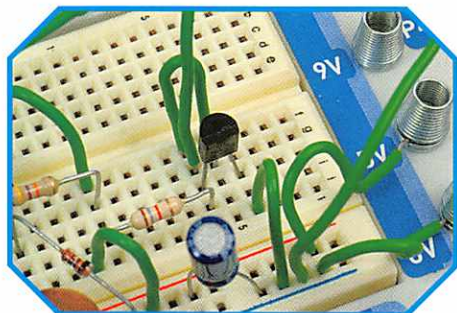
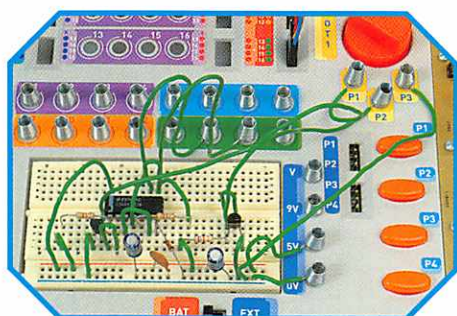
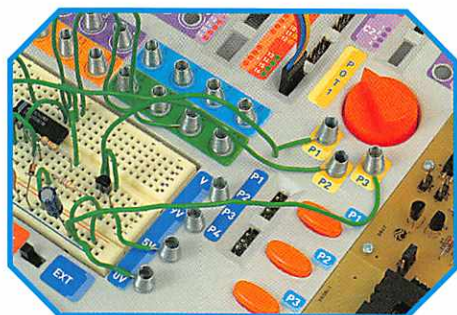
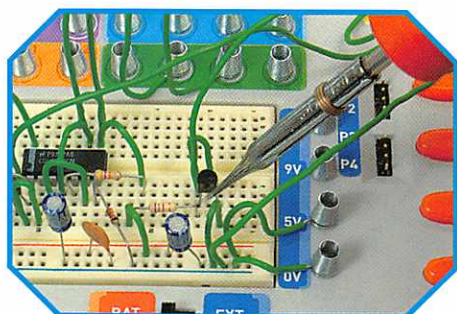
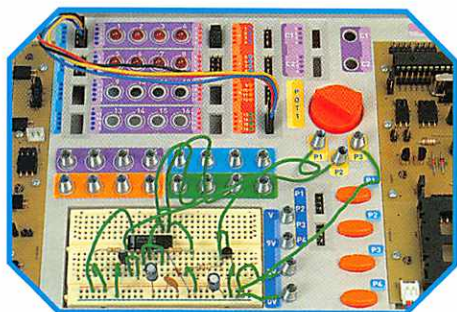
***È** possibile utilizzare un transistor di silicio come sensore di temperatura affidabile ed economico, dato che è molto utile per proteggere i circuiti dall'annoso problema della dissipazione della temperatura generata all'interno, comune a tutti gli strumenti. I circuiti digitali sono particolarmente sensibili a questo problema.*

Il sensore

Osservando attentamente lo schema da sinistra a destra troveremo un transistor, siglato come Q1, con la base e il connettore ponticellati, l'emettitore collegato al negativo dell'alimentazione e il collettore al positivo tramite una resistenza da 1K8. La caduta di tensione sul transistor diminuisce di 0,2 mV per ogni grado in più di temperatura, sfrutteremo questa proprietà per utilizzare il transistor come sensore di temperatura.

Il circuito

La caduta di tensione sul transistor impiegato come sensore si utilizza per generare una corrente che attraversa il potenziometro POT1, l'uscita di questo potenziometro serve per ge-

Componenti sulla scheda Bread Board.

Cablaggio della scheda Bread Board.

Schema del circuito.

Il transistor Q1 si utilizza come sensore.

Cablaggio alle molle di contatto.

Con POT1 si regola il punto di attivazione.

Applicazione di calore per provare il circuito.

Esperimento completato.

nerare una corrente di base su un secondo transistor che si utilizza come amplificatore di corrente, generando una caduta di tensione che modifica il valore logico all'ingresso della porta U1A. Questa porta controlla l'attivazione dell'oscillatore costruito con la porta U1B, la resistenza R3 e il condensatore C3, e la sua uscita quando è a livello basso, serve per illuminare un LED, il LED 3. Una terza porta, la U1C, si utilizza come inverter e illumina l'altro LED, il LED 2. Questo circuito ha la particolarità di avere un diodo inserito sul negativo dell'alimentazione dell'integrato per facilitare il funzionamento del circuito.

Quando la temperatura scende sotto al livello regolato con POT1 la tensione sul potenziometro aumenta, quindi aumenta anche la corrente che circola sulla base del transistor Q2, e quando questa è sufficiente a far saturare il transistor, la tensione sull'ingresso della porta U1 è interpretata come livello basso e la sua uscita passa quindi a livello alto; a partire da questo istante l'oscillatore si avvia e i due LED si illuminano in modo intermittente, mentre prima se ne illuminava solamente uno.

Montaggio

Il montaggio si esegue come d'abitudine, facendo attenzione alla polarità dei condensatori elettrolitici e a quella del LED1, che si inserisce sull'alimentazione negativa dell'integrato.

L'esperimento

Con il circuito a temperatura ambiente si regola il potenziometro poco a poco, fino a quando inizieranno a illuminarsi i LED in modo intermittente, a questo punto applicheremo calore al transistor sensore avvicinando la punta del saldatore. Quando il transistor rileva un aumento della temperatura cessa l'intermittenza e solamente uno dei due LED è illuminato, allontaneremo quindi la sorgente di calore e, quando il transistor si sarà raffreddato, inizierà di nuovo l'intermittenza.

LISTA DEI COMPONENTI

U1	Circuito integrato 4093
R1	Resistenza 1K8 (marrone, grigio, rosso)
R2,R3	Resistenza 47 K (giallo, viola, arancio)
C1,C3	Condensatore 10 µF elettrolitico
C2	Condensatore 100 nF
D1	Diodo 1N4148
Q1, Q2	Transistor BC548 o BC547

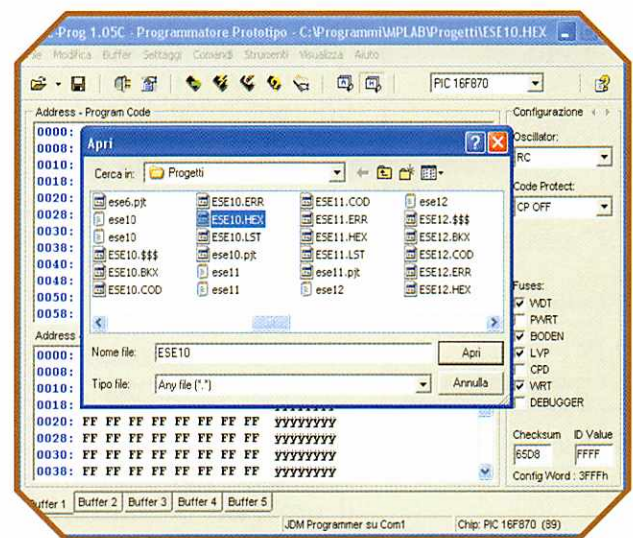
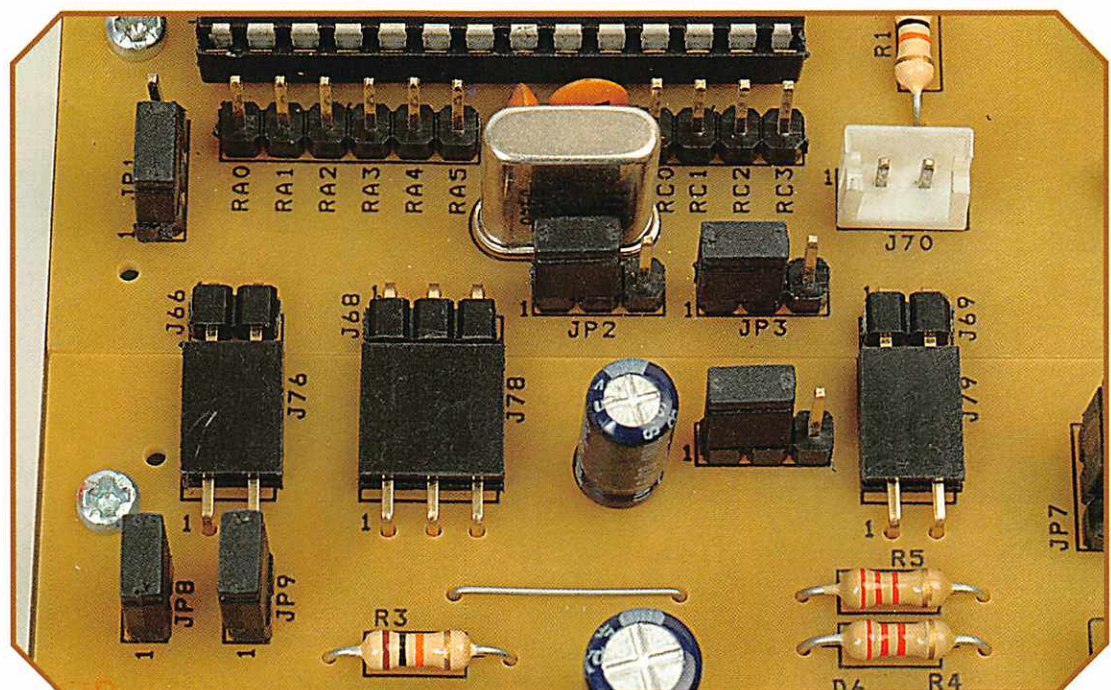
Esercizi 10 e 11: la pratica

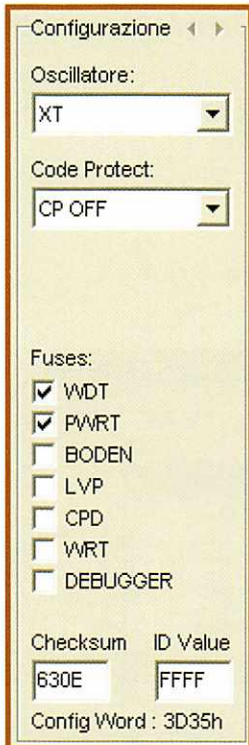
Dopo aver verificato che gli esercizi 10 e 11 vengono compilati correttamente, si esegue la simulazione. Il passaggio finale è quello di scrivere il microcontroller ed eseguire il montaggio dell'hardware necessario perché possa funzionare realmente sul laboratorio.

Scrittura del microcontroller

Per scrivere qualsiasi programma sul microcontroller dobbiamo collegare il laboratorio al PC e utilizzare il software IC-Prog. A questo scopo non abbiamo bisogno di alimentare il laboratorio.

Per rendere effettivo il trasferimento dei dati è necessario configurare l'hardware in modo adeguato. Ricordate che per poter utilizzare l'alimentazione della porta seriale del PC, i ponticelli dei connettori JP1, JP2 e JP3 devono essere nella posizione 1 e 2 e per poter trasferire dati al PIC i ponticelli dei connettori JP8 e JP9 devono essere inseriti.

Eseguita la configurazione hardware, avvieremo IC-Prog e interagiremo col microcontroller. Vi consigliamo di leggere il contenuto del

Aprire il file .hex che si desidera caricare sul PIC.

Ogni volta che si vuole scrivere sul PIC sarà necessario avere la stessa configurazione dei ponticelli.

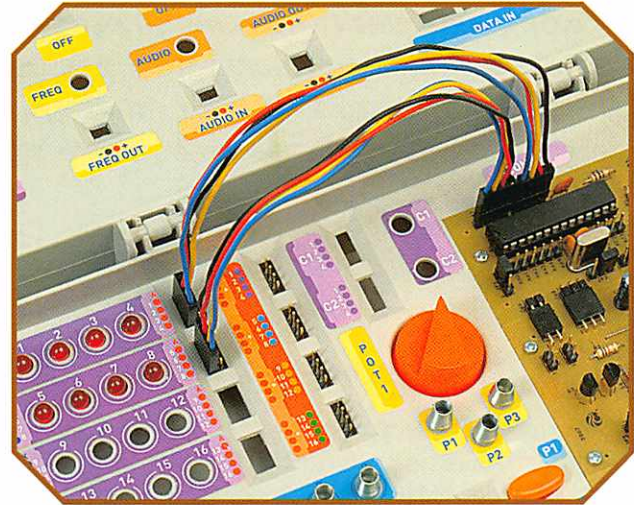
Configurazione
Oscillatore: XT
Code Protect: CP OFF
Fuses:
☑ WDT
☑ PWRT
☐ BODEN
☐ LVP
☐ CPD
☐ WRT
☐ DEBUGGER
Checksum 630E
ID Value FFFF
Config Word : 3D35h

Configurazione di IC-Prog per la scrittura.

dispositivo prima di effettuare un'operazione di cancellazione o di scrittura, nel caso il microcontroller contenesse dei programmi che vogliamo conservare. Potremo quindi cancellare il PIC, leggendone nuovamente il contenuto per verificare la fase di cancellazione. Sarà necessario aprire il file che si desidera scrivere (estensione ".hex") e configurare l'oscillatore e i bit della parola di configurazione, supponendo che sia il dispositivo che la protezione del codice siano correttamente selezionati (PIC16F870 e CP OFF). Attivate le opzioni WDT e PWRT della parola di configurazione e procedete a scrivere il PIC, verificando il processo con una lettura successiva del dispositivo. Il processo di scrittura è sempre lo stesso per tutti i programmi, con qualche differenza per i programmi che richiedono l'attivazione di qualche altro bit della parola di configurazione in funzione dei dispositivi utilizzati, come, ad esempio, la memoria FLASH.

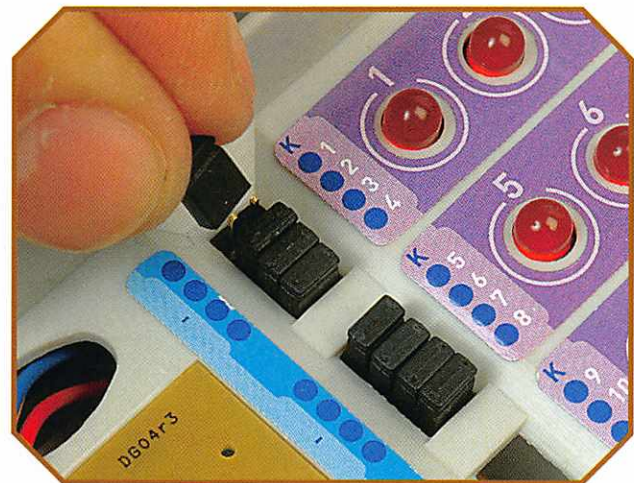
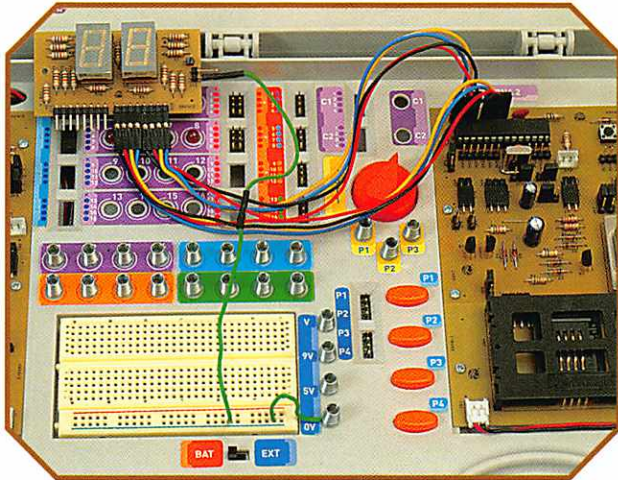
Esercizio 10, la pratica

Caricate l'esercizio 10 sul microcontroller. Questo esercizio provoca l'accensione successiva della barra dei LED ogni 1,3 s, simulando un contatore binario.

Per procedere al montaggio dell'applicazione, fase finale di qualsiasi progetto, ricordate che sarà necessario modificare i ponticelli del laboratorio. I connettori JP1, JP2 e JP3 dovranno avere i ponticelli tra i terminali 2 e 3 per ricevere l'alimentazione dal laboratorio stesso e i connettori JP8 e JP9 dovranno avere i ponticelli estratti.

Nell'esercizio si lavora con la matrice dei LED, che si controlla tramite la porta B del microcontroller. Ricordate che quando si lavora con la matrice dei LED è necessario impostare i ponticelli sui catodi dei diodi LED, come si può vede-

Colleghiamo i terminali della porta B con la matrice dei LED.

re nella figura a fianco. Dobbiamo collegare la porta B con la matrice dei LED tramite due dei cavetti di collegamento a quattro fili. In questo modo uniremo con uno dei cavetti i terminali da RB0 a RB3 con la prima fila dei diodi, e con l'altro i terminali da RB4 a RB7 con la seconda fila della matrice.

Questo montaggio è identico a quello realizzato per l'esercizio 9 in cui si lavorava con il TMR1. Nelle immagini che rappresentavano il montaggio di quella applicazione si utilizzava solamente un cavetto, che collegava i terminali RB0:RB3 con la fila superiore della matrice dei LED, mentre nell'esercizio presente ne stiamo utilizzando due per unire tutta la porta B con la matrice dei LED. Possiamo verificare come i LED

Dobbiamo inserire i ponticelli sui catodi dei diodi LED.

Colleghiamo l'uscita
al display a 7 segmenti.

Colleghiamo un filo all'unico
terminale di uscita che vogliamo utilizzare (RB0).

si accendano in modo sequenziale simulando un contatore binario.

Questo esercizio può anche essere eseguito utilizzando i LED interni a uno dei display. A questo scopo si collega uno dei display a 7 segmenti invece della matrice dei LED, osserveremo così lo stesso risultato ma su un altro dispositivo di visualizzazione. Ricordate che per collegare i display dovremo staccare la scheda DG01 e avvicinarla alla porta B, facendo sempre attenzione a non creare contatti con qualche oggetto metallico. Per quanto riguarda la tensione di riferimento sarà necessario unire il terminale 1 del contatore J13 con 0 V mediante un filo di collegamento. Nell'immagine si

Carichiamo l'esercizio 11 sul PIC.

può vedere il montaggio con il collegamento tra la porta B e il display a 7 segmenti.

Esercizio 11, montaggio

Questo esercizio ha come obiettivo quello di accendere o spegnere il diodo LED collegato a RB0 quando il valore del TMR1 abbia coinciso per 20 volte con un valore precedentemente impostato sui registri CCPR1H/L del modulo CCP.

Sarà necessario configurare nuovamente il laboratorio in modo che i ponticelli siano posti in modo scrittura (JP1, JP2, JP3, JP8 e JP9). Collegate il cavo di comunicazione tra il PC e il laboratorio e fate partire il software IC-Prog. Abituatevi a eseguire sempre gli stessi passaggi: lettura del dispositivo, cancellazione e verifica. Caricate l'esercizio 11 e realizzate la configurazione del software prima di procedere alla scrittura. Cambiate il tipo di oscillatore e il bit di configurazione e programmate il dispositivo.

Dopo aver verificato che il programma sia stato caricato sul PIC potete procedere a riconfigurare il laboratorio, spostando i ponticelli sui connettori JP1, JP2 e JP3, e togliendo i ponticelli di JP8 e JP9.

Il montaggio di questa applicazione risulta più semplice dell'esercizio precedente. In questo caso, infatti, utilizzeremo unicamente un terminale della porta B come uscita, cioè RB0. Dato che si tratta di accendere e spegnere un diodo LED, possiamo fare un semplice montaggio sulla scheda Bread Board in cui porremo una resistenza collegata a massa in serie al dio-

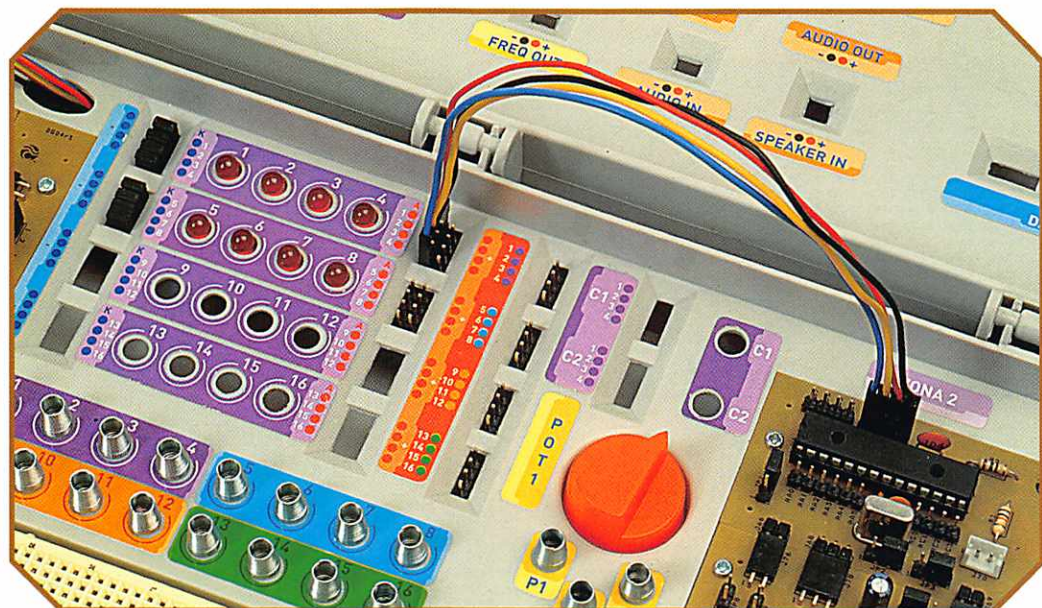
Realizziamo il montaggio sulla scheda Bread Board.

do, che a sua volta va collegato al terminale RB0. Nella figura possiamo vedere questo semplice montaggio.

Un'altra possibilità di realizzare l'applicazione sul laboratorio è utilizzando la matrice dei diodi. Anche se utilizzeremo solamente uno dei diodi, uniremo mediante un cavo di collegamento i terminali RB0:RB3 della porta B con la matrice dei diodi (i ponticelli sui catodi devono essere inseriti).

Come potremo verificare il diodo si accende e si spegne costantemente.

Consigli

Potete vedere come vengono presentate diverse alternative di montaggio per poter in questo modo interagire con i diversi elementi del laboratorio. È altresì interessante trovare le diverse alternative ai codici previsti. Provate come esempio a variare il valore di comparazione di quest'ultimo esercizio e verificate che se si cambia il valore dei registri di comparazione CCPR1H e CCPR1L, varia la frequenza di accensione del diodo LED.

Possiamo anche collegare la matrice dei LED.

Esercizio 11: modulo CCP in modo comparazione, il programma

***I** moduli CCP possono lavorare in tre forme differenti:*
- Acquisizione del valore del TMR1 quando si genera un evento su RC2/CCP1.
- Comparazione del valore del TMR1 con il valore caricato su CCP1H-L; se sono uguali si genera un evento sul pin RC2/CCP1.
- Modulatore di impulsi.
In questo esercizio lavoreremo con questo modulo nel modo comparazione. Questo dispositivo comporta implicitamente di lavorare con il TMR1, utilizzeremo inoltre gli interrupt.

Enunciato

Vogliamo sviluppare un programma che accenda e spenga il LED corrispondente a RB0 quando il valore del temporizzatore TMR1 coincide con un valore prefissato. Questo valore verrà scritto sui registri CCP1H/L e utilizzeremo come esempio: 10101010 10101010. Per poter osservare il lampeggio del LED faremo in modo che si accenda o si spenga ogni 20 interrupt.

Lavoreremo con la porta B per fornire l'uscita al LED, e anche con i dispositivi CCP, TMR1 e interrupt.

Questi esercizi hanno un livello di difficoltà medio, dato che utilizzano diversi dispositivi, ma sono esercizi di base per imparare a gestire questi dispositivi e, in seguito, saperli utilizzare in applicazioni più complesse.

Organigramma dell'applicazione.

Organigramma

Utilizziamo gli organigrammi per fare una pianificazione generale di come risolvere l'esercizio. Questo caso è simile a quelli presentati in precedenza nel senso che dovremo prevedere un programma principale in cui configurare tutti i dispositivi, e una subroutine di trattamento degli interrupt in cui eseguiremo le azioni finali.

Registri da configurare	Registri da inizializzare
TRISB	PORTB
OPTION REG	CCPR1H
PIE1	CCPR1L
CCP1CON	TMR1H
INTCON	TMR1L
T1CON	CONT
	PIR1

Registri con cui vogliamo lavorare.

Nel programma principale configureremo la porta di uscita resettandola per cancellare eventuali valori residui, configureremo il predivisore per il WDT, abiliteremo gli interrupt, configureremo il modulo CCP, il TMR1 e inizializzeremo tutti i registri con cui vogliamo lavorare. Fatto questo non ci resta che attendere all'interno di un ciclo infinito che si producano gli interrupt dovuti al modulo CCP.

Nella subroutine dell'interrupt eseguiamo le azioni sull'uscita solamente se si sono generati 20 interrupt.

```
;
;       ESERCIZIO: MODULI CCP in modo comparazione
;     ----------------------------------------------

;Programma che accende e spegne il LED corrispondente a RB0 quando il valore TMR1
;coincide con il valore che abbiamo preimpostato sui registri CCP1H-L, ad esempio,
;10101010 10101010. Per poter osservare il lampeggio faremo accendere e spegnere
;il LED ogni 20 interrupt.

        LIST    P=16F870         ;Definiamo il nostro PIC
        INCLUDE "P16F870.INC"    ;File di definizione dei registri interni

Cont1           equu    0x20
Cont2           equu    0x21


                ORG     0
                GOTO    INIZIO
                ORG     4
                GOTO    INT
                ORG     5
```

Intestazione del codice.

```
;Programma principale.

INIZIO          clrf    PORTB
                clrf    PORTC            ;Disattiviamo gli I/O
                bsf     STATUS,RP0       ;Passiamo al banco 1
                clrf    TRISB            ;Porta B uscita
                movlw   b'11101111'
                movwf   OPTION_REG       ;Prescaler da 128 per il WDT
                bsf     PIE1,CCP1IE      ;Abilitiamo gli interrupt tramite CCP1
                bcf     STATUS,RP0       ;Torniamo al banco 0
                movlw   b'00001011'
                movwf   CCP1CON          ;Configuriamo il CCP1 in modo comparazione con attivazione
                                         ;speciale (resetta il TMR1) quando coincide con il
                                         ;valore su CCPR1H e CCPR1L
                movlw   b'10101010'
                movwf   CCPR1H
                movlw   b'10101010'
                movwf   CCPR1L           ;Carichiamo i registri CCP1H-L con un valore (43690)
                clrf    TMR1L
                clrf    TMR1H            ;Resettiamo il TMR1
                movlw   b'11000000'
                movwf   INTCON           ;Abilitiamo gli interrupt
                movlw   b'00000001'
                movwf   T1CON            ;Attiviamo il TMR1 e configuriamo il prescaler con 1/1
                movlw   .20
                movwf   Cont1
                bcf     PIR1,TMR1IF      ;Resettiamo il flag del TMR1
                bcf     PIR1,CCP1IF      ;Resettiamo il flag del modulo CCP1

NIENTE          clrwdt
                goto    NIENTE           ;Ciclo che non fa niente sino a quando
                                         ;non si produce l'interrupt
                END
```

Esempio di codice che formerà il programma principale.

Codice

Inizieremo a sviluppare il codice intestando il programma con i commenti. Definiremo il microcontroller e la libreria dei registri e, mediante le direttive ORG, le zone di memoria che occuperà il programma.

Anche se è consigliabile definire le variabili man mano che si rendono necessarie, lo si può fare all'inizio del programma facendo una stima di ciò di cui avremo bisogno. Sappiamo che ne utilizzeremo un paio per il contatore, ma ancora non sappiamo se avremo bisogno di altre variabili, quindi ne creeremo due. Si possono creare quante variabili si vuole, ma bisogna tener presente che occupano memoria e le risorse utilizzate devono essere sempre ottimizzate. Nella figura possiamo vedere la forma che sta assumendo il codice e come abbiamo definito due variabili, anche se per ora non sappiamo se le utilizzeremo.

Programma principale

Mettiamo l'etichetta "Inizio" e iniziamo a programmare. Cancelliamo eventuali valori residui che ci potrebbero essere sulle porte (sarebbe consigliabile farlo solamente sulle porte che desideriamo utilizzare). Configuriamo la porta B come uscita cancellando il registro TRISB. Configuriamo il Watchdog modificando il registro OPTION_REG. Abilitiamo gli interrupt dovuti al modulo CCP e configuriamo quest'ultimo per farlo lavorare nel modo comparazione e che imposti a 0 il temporizzatore se è stato raggiunto il valore desiderato. Il registro CCPCON permette di configurare il modulo CCP. Dobbiamo inizializzare tutti i re-

```
;Routine di trattamento dell'interrupt.

INT             decfsz  Cont1,F         ;Verifichiamo se il tempo è trascorso
                goto    CONTINUA        ;Ancora no
                movfw   PORTB           ;Se è spento si accende
                xorlw   b'00000001'     ;Altrimenti si spegne
                movwf   PORTB
                movlw   .20
                movwf   Cont1           ;Reinizializziamo il Cont1
CONTINUA        bcf     PIR1,TMR1IF     ;Azzeriamo il flag del TMR1
                bcf     PIR1,CCP1IF     ;Azzeriamo il flag del modulo CCP1
                clrf    TMR1L
                clrf    TMR1H           ;Resettiamo il TMR1
                retfie
```

Codice della subroutine dedicata all'interrupt.

```
Build Results
Building ESE11.HEX...

Compiling ESE11.ASM:
Command line: "D:\PROGRA~1\MPLAB\MPASMWIN.EXE /p16F870 /q C:\PROGRA~1\MPLAB\PROGET~1\ESE11.ASM"
Message[302] C:\PROGRA~1\MPLAB\PROGET~1\ESE11.ASM 46 : Register in operand not in bank 0.  Ensur
Message[302] C:\PROGRA~1\MPLAB\PROGET~1\ESE11.ASM 48 : Register in operand not in bank 0.  Ensur
Message[302] C:\PROGRA~1\MPLAB\PROGET~1\ESE11.ASM 49 : Register in operand not in bank 0.  Ensur

Build completed successfully.
```

Risultato della compilazione.

gistri, quindi su CCPR1H/L caricheremo il valore desiderato, resetteremo i registri TMR1H/L, inizializzeremo la variabile che utilizzeremo come contatore e resetteremo i flag di segnalazione riferiti ai dispositivi con cui vogliamo lavorare. In ultimo configureremo il registro INTCON per abilitare gli interrupt e il registro T1CON del temporizzatore.

Ora non ci resta che attendere all'interno di un ciclo che si generino gli interrupt. Per maggior sicurezza, per evitare che il programma possa rimanere bloccato, resettiamo il WDT (Watchdog) all'interno del ciclo.

Per aggiungere registri da visualizzare selezioneremo Add.

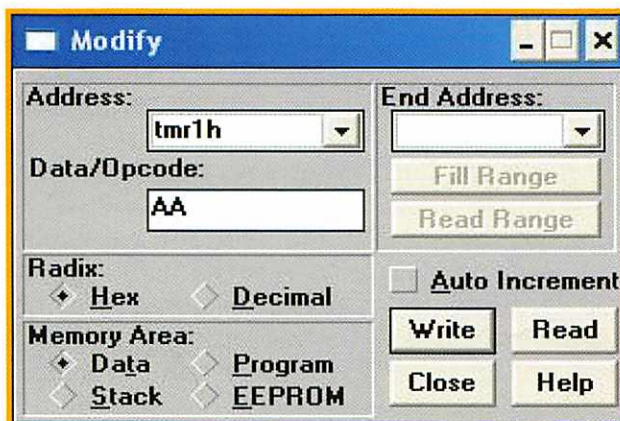
Routine di servizio dell'interrupt

Questa routine è simile a quella progettata negli esercizi precedenti. Quando si genera un interrupt decrementiamo il contatore e, se questi è arrivato a 0, agiamo sull'uscita, inizializiamo il valore del contatore e quello del temporizzatore, e resettiamo i flag di indicazzione. Nel caso in cui il contatore non sia ancora a 0 resetteremo unicamente i flag e inizializzeremo il valore del temporizzatore.

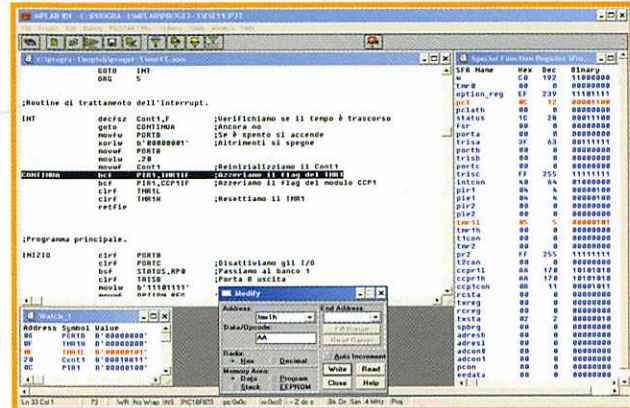
Compilazione

Abbiamo progettato il nostro codice e lo dobbiamo compilare. Creiamo un progetto, associamo il codice creato a quest'ultimo e selezioniamo le opzioni di compilazione (Build All). Per l'esempio che abbiamo presentato il risultato di questo processo di compilazione è quello mostrato nella figura.

Questo esempio lo potrete trovare sul secondo CD-ROM con il nome "ese11.asm", vi consigliamo comunque di provare a realizzare voi stessi i programmi, e di tenere questi esempi solamente come fonte di consultazione.

Cambiamo il formato dei dati presentandoli in binario.

Forziamo il valore del registro TMR1H.

Simulazione

Per simulare il corretto funzionamento del programma apriremo le finestre abituali, una per i registri delle funzioni speciali e l'altra finestra dedicata ai registri che vogliamo vedere in modo indipendente. In questa finestra selezioneremo i registri: PORTB, TMR1H, TMR1L, CONT1 e PIR1, presenteremo i dati in formato binario. Ricordate che per fare questo è necessario cliccare con il pulsante sinistro del mouse sul simbolo situato nell'angolo superiore sinistro della finestra e selezionare Edit Watch. A questo punto si apre una finestra dove possiamo vedere i registri che vogliamo visualizzare, questi possono essere modificati selezionandoli uno per uno premendo Properties. Apparirà una nuova finestra come quella riportata nella figura a fianco, in cui sceglieremo l'opzione Binary nella sezione Format.

Aspetto generale di MPLAB durante la simulazione.

Quando eseguiamo passo a passo la simulazione vediamo come si configurano correttamente i dispositivi nel programma principale, come al momento in cui arriva l'attivazione dell'interrupt entri nella subroutine e terminata questa ritorni al programma principale, continuando con la configurazione e inizializzazione dei registri. All'interno del ciclo infinito, premendo F7 (simulazione passo a passo) incrementeremo di una unità per volta i registri del temporizzatore, quindi per non perdere troppo tempo e arrivare al valore desiderato, forzeremo il valore del registro TMR1H a 10101010. Faremo questo con la finestra Modify, inserendo il valore nel suo equivalente esadecimale (10101010=AA). Continuiamo a eseguire passo a passo, fino a quando il temporizzatore raggiungerà il valore desiderato, momento in cui si genera un interrupt, cambia lo stato del registro PIR1 e saltiamo alla routine di servizio per l'interrupt.

Nella subroutine si eseguono tutti i passi correttamente, ma se vogliamo vedere come funziona il codice sull'uscita dobbiamo forzare anche il valore del registro CONT1. Se scriviamo su questo registro il valore 01h la prossima volta che entreremo nella subroutine agiremo sull'uscita. In questo modo il programma rimane simulato in modo soddisfacente, ed è pronto per essere scritto sul PIC.